miniChemistry Documentation

This guide provides a high level overview of the project, its core packages and a brief reference of

Project Structure

‘miniChemistry‘ is organised into two major cores:

* **Core** contains modelling primitives for chemical substances and reactions. It is responsible fo
* **Computations** builds on top of ‘Core‘ and performs stoichiometric calculations using linear alg

The repository also includes a small **Utilities** package with helper functions and classes, an ‘EX

Core subpackages

‘‘‘
Core/
 Database/   # CSV databases (solubility tables, activity series)
 ReactionMechanisms/   # Functions that produce reaction products
 Tools/   # Parser, reaction predictor and helpers
 Substances.py   # Particle, Simple, Ion and Molecule classes
 Reaction.py   # Reaction class describing reagents and products
 CoreExceptions/   # Custom exceptions
‘‘‘

Computations subpackages

‘‘‘
Computations/
 ReactionCalculator.py   # Main class for stoichiometric calculations
 SSDatum.py   # Datum with linked substance information
 ComputationExceptions/   # Exceptions for calculation modules
 CalculatorFiles/   # Text files with formulas and units
‘‘‘

Both cores rely on a small ‘Utilities‘ package that provides input validation helpers and a tiny ‘File‘ a

Relations between cores

The ‘Computations‘ core uses the ‘Reaction‘ class and the substance classes from the ‘Core‘ packa

‘‘‘
[ReactionCalculator] --uses--> [Reaction]
[Reaction] --has--> [Molecule] / [Simple] / [Ion]
[Molecule] and [Simple] inherit from [Particle]
‘‘‘

UML diagrams

Core

‘‘‘
class Particle <<abstract>>
    +composition: Dict[Element, int]
    +charge: int
    +from_string(...)
    +formula()
‘‘‘